Task 5

Effects of Residual Networks

Alexander Garcia

May 2025

Contents

1 Introduction

The goal of this phase is to test the effects of using residual and pretrained networks such as ResNet, VGG, etc. Residual networks add inputs into their outputs. The trick here is keeping track of the shape. Conv with increasing filters and MaxPooling change shapes so the implementation has to be adjusted slightly in those situations. Pretrained networks involve utilizing a model that has already been trained and then fine-tuning it to your data set. When doing this you can alter any layers parameters to be trainable or not.

2 Effects of Residual Connections

When using residual connections the graph of a model will look a bit different. Instead of being perfectly linear there are now some extra branches each time I utilize the "add" method from layers. A graph of residual connections looks like the image below.

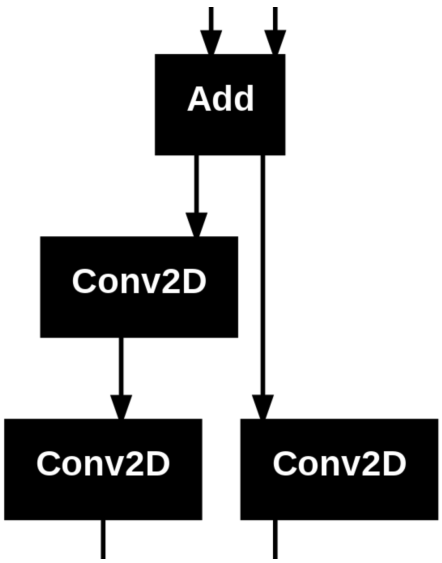
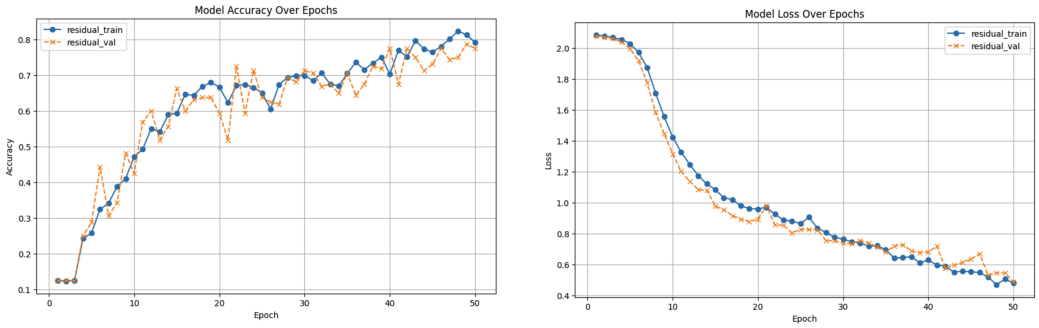


Figure 1: Residual Connections

In this attempt I noticed the loss curve was smooth and the training did not reach early stopping by the 50th epoch. However, I still cannot get rid of the spikes in validation or training accuracy. Overall, the model ran for

about 1 hour with most epochs (all except the first) taking approximately 46 seconds on the T4 GPU on Google Colab. The validation accuracy peaks at just about 80% however it should be noted this is not the 5 layer multi-million parameter model. This initial test was to gauge the spikes and if they would subside since validation and test accuracy are already doing well. When time permits, another model with residual connections will be run on the 5 layer but given it will need well over 1 hour, that has to wait.

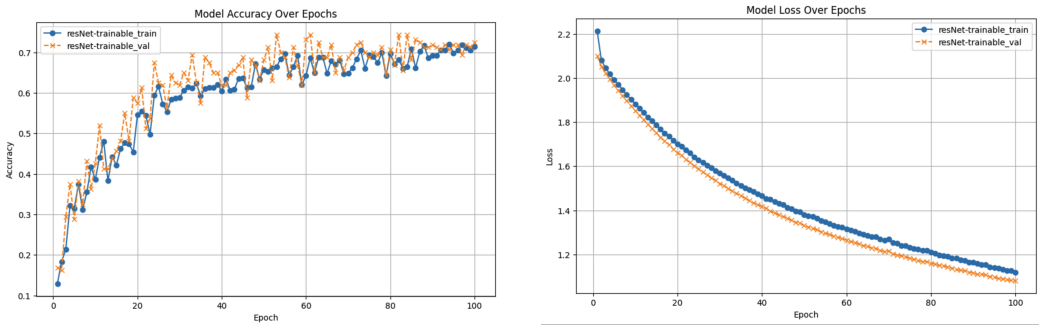


(a) Accuracy (b) Loss

Figure 2: Residual connection metrics

3 Fine Tune

In this section I tried utilizing the ResNet50 from Keras. I ran 3 different tests - one in which no layers were trainable and then two with different amounts of ResNet layers being trainable. I noticed very quickly the parameters will skyrocket if even a couple layers are made trainable and the model overfits quickly. The loss curve is incredibly smooth but the validation spikes seem to be more frequent than previous runs. They do taper out toward the end of the model but overall its still not great. The model also converges very slowly. The final model tested ran for 100 epochs and did not reach early stopping at that time while achieving only 73% validation accuracy.



(a) Accuracy (b) Loss

Figure 3: ResNet metrics

4 Conclusion

In summary, I attempted to utilize residual connections and model fine-tuning to increase validation accuracy and reduce the noise while training. Validation accuracy in both the residual connections and ResNet show promise of reaching greater than 90% if they ran for long enough, but neither could reduce the spikes. I found it interesting the loss function was much smoother but the accuracy was still relatively unstable throughout all the tasks completed thus far.